BERT

What is BERT?

- Bidirectional Encoder Representations from Transformers. It is a pre-trained model designed for learning bidirectional representations from unlabeled data to get the left and right context of the text.
- So what do we mean by a pre-trained model? In today's scenario, the biggest challenge for natural language processing(NLP) is shortage of training data. NLP is a wide field consisting of different tasks for which the dataset consists of only a few hundred or a few thousand human labeled training examples.To bridge this gap , researchers have found a way for training general purpose language representation models using the enormous amount of unannotated text on the web. This is called pre-training and this can then be fine tuned on specific NLP tasks like sentiment analysis,question answer chatbots , resulting in improved accuracy.
- Now the pre-trained representations can either be context free or contextual . Contextual representations are further classified as unidirectional or bidirectional. Context free models generate single word embeddings for each word in the text whereas the contextual model generates embedding of each word based on the other words in the text. Now to generate such embeddings it can look only from left to right or right to left or both.
- How is bidirectional powerful? Bidirectionality can not be trained by simply conditioning each word on its previous and next words , since this allows the model to "peak" the answer .So to solve this we use masked language models inspired by the cloze task

Applying Pre-trained Language Representations

There are two existing strategies for applying downstream tasks:

1. Feature Based Approach: Fixed features which are context sensitive are extracted from the pretrained model
2. Fine Tuning Approach: It needs minimum task specific parameters and is trained on the downstream tasks by fine tuning all pre trained parameters

Where is BERT used?

BERT has a wide range of applications :

1. Text Summarisation
2. BioBERT
3. FinBERT
4. SciBERT
5. Google Smart Search
6. Protein Language Modeling
7. Question-Answering chatbots

Why do we need BERT?

1. BERT has been pre-trained on a very large training set , one of them being wikipedia .It can then be fine tuned for some NLP specific tasks having small datasets and still have good performance.
2. BERT accounts for the context of a word. Here the word “right” is being used in different contexts and BERT will return different embeddings for different uses of the same word.

    - *I’m sure I’m **right**.*
    - *Take a **right** turn at the intersection.*

3. BERT is open source and accessible to all.

How does BERT work?

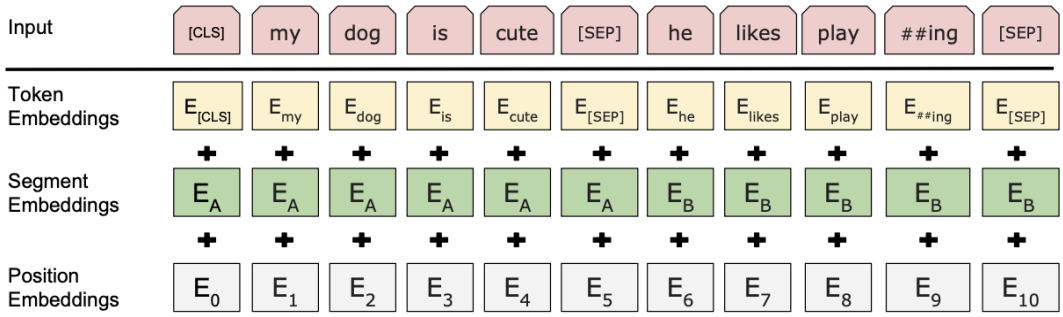
BERT model architecture is a multilayer bidirectional transformer encoder. The transformer uses an important mechanism called attention mechanism where it is able to learn relations between words in textual data. Also since BERT is a language representation model we do not need the decoder.

Dataset

For pre-training the model , BookCorpus consisting of 800M words and English wikipedia consisting of 2500M words was used. Any tables ,lists and headers were not considered and only text passages were used.

Input Representations

Similar to transformers , the text data needs to be preprocessed , so that the input to the BERT model is input embedding which is a sum of the token embeddings , the segment embeddings and the position embeddings.



Here a “sentence” refers to a span of contiguous text and “sequence” refers to input token sequence BERT which can be one / two sentences packed.

1. Token embeddings: The first token of sequence is a start token [CLS] and to separate between two sentences we add a special token [SEP].
2. Segment embeddings: It is an indicator showing which sentence the tokens belong to.
3. Position embeddings: Since it is a transformer architecture , it doesn't take word step by step or sequentially .It is kind of hard for a model to make out how far apart or close the two tokens are from each other. So we have position embeddings.

Pre-training BERT

BERT is pre-trained using two unsupervised tasks:

Masked Language Model

The masked language model randomly masks 15 % of the tokens from the input and the aim is to predict the original vocabulary id of the masked token based only on its context. This enables the representation to contain both the left and the right context to train the model.

Input: The man went to the $[MASK]_1$. He bought a $[MASK]_2$ of milk .
Labels: $[MASK]_1$ = store; $[MASK]_2$ = gallon

This can also help to model the relationships between sentences by pre-training on a very simple task that can be generated from any text corpus.

This gives a bidirectional model but the downfall is that there is a mismatch between the pre-training and fine-tuning as the [MASK] token does not appear in fine-tuning. To resolve this , the training data generator choses 15% of the token randomly . Now if the i^{th} token is chosen then we replace i^{th} token with [MASK] token 80% of the time , random token 10% of the time and unchanged i^{th} token 10% of the time.

Next Sentence Prediction

In order to train a model that understands the relationship between sentences which is directly not captured by language modeling.
This can be trivially generated from a monolingual corpus with balanced training examples of actual next sentence(“IsNext”) and not next sentence(“NotNext”).

Input = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]

penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

Fine tuning BERT

The pre-trained BERT model can then be used for several downstream tasks such as question answering , paraphrasing etc. The fine tuning task is relatively inexpensive and can be done by adding just a single layer on top of the core model.

BERT Limitations

I recently read a paper titled “What BERT is not” where the author applied some diagnostics and found that BERT struggles with challenging inferences and role based event prediction.

- Commonsense and pragmatic inference

In the figure below , there is a certain type of pragmatic inference that humans can easily pick up on. BERT was able to prefer good completion(“lipstick”) over bad completions(“mascara/bracelet”) but not with a good confidence score . Therefore it failed at grasping the context provided in the sentence to a higher level.

Context	Expected	Inappropriate
He complained that after she kissed him, he couldn't get the red color off his face. He finally just asked her to stop wearing that ____	*lipstick*	*mascara* | *bracelet*
He caught the pass and scored another touchdown. There was nothing he enjoyed more than a good game of ____	*football*	*baseball* | *monopoly*

- Negation

Context	$\text{BERT}_{\text{LARGE}}$ predictions
A robin is a ____	*bird, robin, person, hunter, pigeon*
A daisy is a ____	*daisy, rose, flower, berry, tree*
A hammer is a ____	*hammer, tool, weapon, nail, device*
A hammer is an ____	*object, instrument, axe, implement, explosive*
A robin is not a ____	*robin, bird, penguin, man, fly*
A daisy is not a ____	*daisy, rose, flower, lily, cherry*
A hammer is not a ____	*hammer, weapon, tool, gun, rock*
A hammer is not an ____	*object, instrument, axe, animal, artifact*

Table 13: NEG-88-SIMP predictions by $\text{BERT}_{\text{LARGE}}$

BERT was good at positive inference but not good at negative inference . This is something which is also observed in the case of protein language modeling where BERT is not able to predict structure well for a bad training example.